CSS385: Assignments 2 and 3 Question Sheet

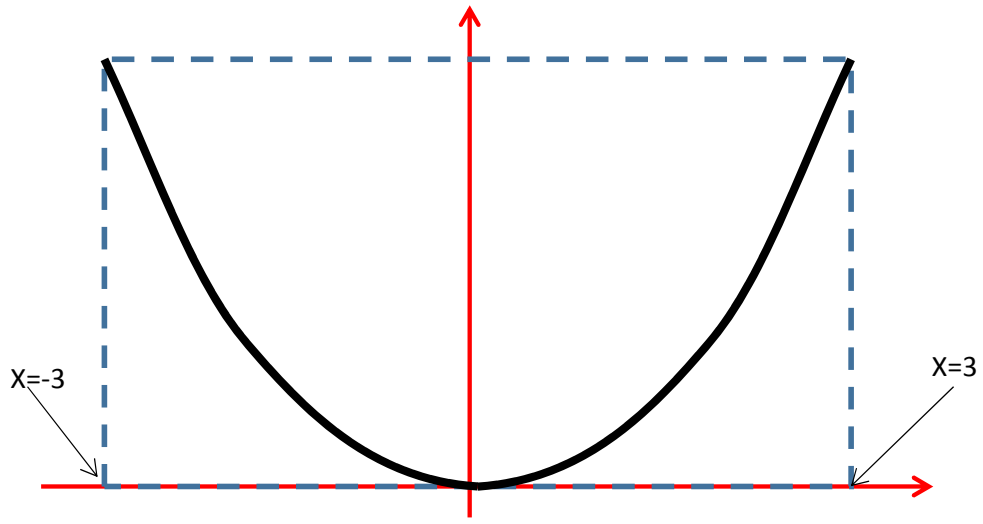
Instructions:

- In questions where you are asked to explain, please be concise.
- Show your work when necessary, be neat, precise, and brief!
- To help us grade your assignments and return this to you in a timely fashion please:
  - Put your name and answers in the answer sheet only (separate link provided). Anything you write outside of the answer sheet **will not** be graded.
  - Provide your answers in the order of the problems.
  - Please use only one side of 8.5x11 paper.
  - Please make sure you bring a hardcopy print out **of the answer sheet (!!only!!)** to submit at the beginning of class. Please do not print out the problems.

Your assignment may not be graded if any of the above is violated, you have been warned.

Assignment 2

1. (**12pt**) If you want your patrol to travels along the function $y = 4x^2$ in the range of $-3 \leq x \leq 3$. In other words, you want to fit the following plot:



into your game window and have the patrol follow the parabola. The following function plots the patrol path as a sequence of circles:

```
private void PlotCurve()
{
    for (int x = 0; x < World.WorldDimension.X; x++)
    {
        float y = GetYValue(x);
        XNACS1Circle c = new XNACS1Circle(new Vector2(x, y), 0.2f);
        mPlotSet.AddToSet(c);
    }
}
```

Where *GetYValue()* is defined as

```
private float GetYValue(float x)
{
    compute the value for useX

    compute the value for yScale

    return yScale * 4 * (useX * useX);
}
```

Now, consider the *GetYValue(float x)* function as given above, and the fact that the *GetYValue()* is only called from the *PlotCurve()* function:

a. (**4pts**) Minimum values:

i. (**1pt**) What is the smallest value that will be passed into the *GetYValue()* function (smallest number for x, the parameter of the *GetYValue()* function)?

0

ii. (**2pt**) In this case, what should be the computed value for ***useX*** ?

-3

iii. (**1pt**) In this case, what should be the return value from *GetYValue()*?

World.WorldDimension.Y

b. (**4pts**) Maximum values:

i. (**1pt**) What is the largest value that will be passed into the *GetYValue()* function (largest number for x, the parameter of the *GetYValue()* function)?

World.WorldDimension.X-1

(OK to say: World.WorldDimension.X)

ii. (**2pts**) In this case, what should be the computed value for ***useX***?

3

iii. (**1pt**) In this case, what should be the return value from *GetYValue()*?

World.WorldDimension.Y

c. (**2pt**) Consider your answer to a and b, derive the expression for computing ***yScale***.

World.WorldDimension.Y/36

d. (2pt) Now derive the expression for computing ***useX***.

$$\frac{6x}{World.WorldDimension.X} - 3$$

Assignment 3

1. **(5pt)** Give the following code that implements home-in chasing:

```
91          private void ComputeNewDirection(Vector2 toTarget)
92          {
93              double cosTheta = Vector2.Dot(toTarget, FrontDirection);
94              float theta = MathHelper.ToDegrees((float)Math.Acos(cosTheta));
95              if (theta > 0.001f)
96              {
97                  Vector3 frontDir3 = new Vector3(FrontDirection, 0f);
98                  Vector3 toTarget3 = new Vector3(toTarget, 0f);
99                  Vector3 zDir = Vector3.Cross(frontDir3, toTarget3);
100                 RotateAngle += Math.Sign(zDir.Z) * 0.15f * theta;
101                 VelocityDirection = FrontDirection;
102             }
103         }
```

Please identify by indicating the exact line number(s) where we are performing the following tasks. In cases where the specified task is not supported by the given code, please just say "***Not Supported***". For example, for the task:

Compute the color of the chaser. **Answer: *Not Supported***

a. **(1pt)** Compute the vector from the chaser to the target.
Not Supported

b. **(1pt)** Compute the distance between the chaser and the target.
Not Supported

c. **(1pt)** Compute the angle (in degrees) between the chaser's front direction and the direction towards the target.
Line: 93 or 94 (both are ok)

d. **(1pt)** Check to ensure the chaser does not overlap the target.
Not Supported

e. **(1pt)** Changing the front direction of the chaser.
Line: 100
(NOT 101: RotateAngle changes FrontDirection, VelocityDireciton has nothing to do with FrontDirection)

2. **(4pt)** Continue with the above question:
    a. **(2pt)** Does the given method belongs to the **chaser-class** or the target-class?
    b. **(2pt)** At what rate does the given method turns?

15% of the current angle. Just say 15% is ok. or 0.15 of the angle is also ok

The important point is %

3. **(3pt)** Our hero has the following states:
    - Run
    - Walk
    - Attack
    - Stand
    - Sit

We want to be able to transition the hero into each of the states, and the hero should never get *stuck* in any of the states (the hero can always transit out of a state), ***not*** counting remaining in the same state as a transition, and note that a transition from state-A to state-B is ***different*** from the transition from state-B to state-A:

a. **(1pt)** How many distinct state transitions can we have in minimum?

5 (or n): Each state has 1 distinct transition to another state

b. **(2pt)** How many distinct state transitions can we have in maximum?

5*4 (or n * (n-1)): Each state has 4 transitions to each other state, and there are 5 distinct states.

---

4. (**2pt)** Given the following vectors:
    - $\overrightarrow{V_1} = \begin{bmatrix} 10 \\ 4 \end{bmatrix}$
    - $\overrightarrow{V_2} = \begin{bmatrix} -2 \\ 5 \end{bmatrix}$
    a. **(1pt)** What is the value of $\overrightarrow{V_1} \cdot \overrightarrow{V_2}$?

0

b. **(1pt)** From **a**, what do we know about these two vectors?

Given vectors are perpendicular, or angle between the two vectors is 90-degrees